

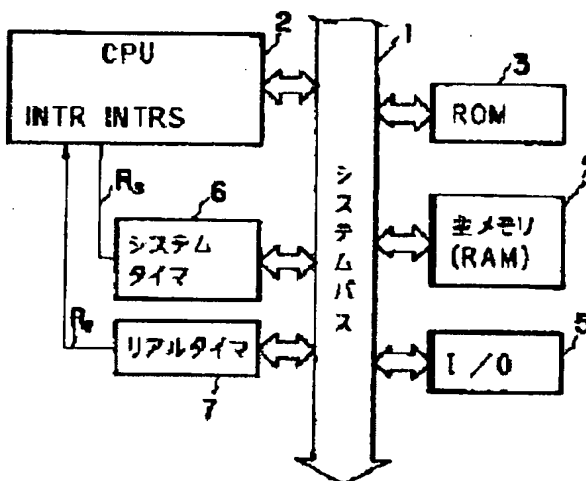
## TASK WAIT TIME MANAGEMENT DEVICE FOR COMPUTER SYSTEM

Patent number: JP7152588  
Publication date: 1995-06-16  
Inventor: NAKAMURA MASAYOSHI  
Applicant: TOKYO SHIBAURA ELECTRIC CO  
Classification:  
- International: G06F9/46; G06F9/46; (IPC1-7): G06F9/46  
- european:  
Application number: JP19930301915 19931201  
Priority number(s): JP19930301915 19931201

Report a data error here

### Abstract of JP7152588

**PURPOSE:** To start a pertinent task at an accurate timing in response to the task execution request, which requests the start after elapse of a designated wait time  $T_s$ , in a computer system. **CONSTITUTION:** A system timer 6 which generates a periodical interrupt request  $R_s$  with a prescribed period  $T_o$  and a real timer 7 which counts a fractional time  $T_r$  from the present time to the occurrence time of the next periodical interrupt request  $R_s$  of the system timer 6 and generates a count end interrupt request  $R_r$  at the time of the end of counting of a time ( $t$ ) set to a real timer queue 8 are provided, and the time obtained by subtracting the fractional time  $T_r$  from a designated wait time  $T_s$  is divided by the prescribed period  $T_o$  to obtain a quotient  $N$  and a remainder time ( $n$ ), and the quotient  $N$  is set to a system timer queue 9, and the remainder time ( $n$ ) of a pertinent task is set to the real timer queue 8 in accordance with the occurrence of the periodical interrupt request  $R_s$  after elapse of prescribed periods whose number corresponds to this set number  $N$  of stacks (quotient  $N$ ). The pertinent task is started in accordance with the occurrence of the count end interrupt request  $R_r$  of the real timer 7.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-152588

(43) 公開日 平成7年(1995)6月16日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0 B	8120-5B		
	C	8120-5B		

審査請求 未請求 請求項の数 1 O L (全 7 頁)

(21) 出願番号 特願平5-301915

(22) 出願日 平成5年(1993)12月1日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 中村 正義

東京都府中市東芝町1番地 株式会社東芝

府中工場内

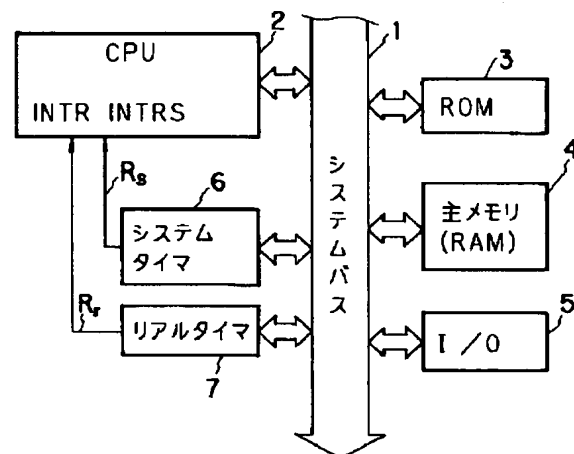
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 計算機システムのタスク待ち時間管理装置

(57) 【要約】

【目的】 計算機システムにおいて、指定待ち時間  $T_s$  経過後に起動させるタスク実行要求に対して、正確なタイミングで該タスクを起動させる。

【構成】 規定周期  $T_o$  で定期割込要求  $R_s$  を発生するシステムタイマ6と、現在時刻からシステムタイマ6の次の定期割込要求  $R_s$  発生時刻までの端数時間  $T_r$  を計時すると共に、リアルタイマ待ち行列8に設定された時間  $t$  の計時が終了すると計時終了割込要求  $R_r$  を発生するリアルタイマ7とを設け、指定待ち時間  $T_s$  から端数時間  $T_r$  を差引いた時間を規定周期  $T_o$  で除算して商  $N$  と余り時間  $n$  を求め、商  $N$  をシステムタイマ待ち行列9に設定し、この設定されたスタック数  $N$  (商  $N$ ) に対応する数の規定周期経過後の定期割込要求  $R_s$  発生に応じて、該タスクの余り時間  $n$  をリアルタイマ待ち行列8に設定する。そして、リアルタイマ7の計時終了割込要求  $R_r$  発生に応動して該タスクを起動する。



(2)

特開平 7-152588

1

## 【特許請求の範囲】

【請求項 1】 現在時刻から指定待ち時間経過後に起動させるタスク実行要求を待ち行列に設定して、前記指定待ち時間経過後に該当タスクを起動させる計算機システムのタスク待ち時間管理装置において、

規定周期で定期割込要求を発生するシステムタイマと、現在時刻から前記システムタイマの次の定期割込要求発生時刻までの端数時間を計時すると共に、リアルタイム待ち行列に設定された時間の計時が終了すると計時終了割込要求を発生するリアルタイムと、

前記待ち時間を指定したタスク実行要求に応じて、この指定待ち時間から前記端数時間を差引いた時間を前記規定周期で除算して商と余り時間を求める除算手段と、

この除算手段にて得られた商を該当タスクの待ちスタック数としてシステムタイマ待ち行列に設定するシステムタイマ待ち行列設定手段と、

このシステムタイマ待ち行列に設定されたスタック数に対応する数の規定周期経過後の定期割込要求発生に応じて、該当タスクの前記余り時間を前記リアルタイム待ち行列に設定するリアルタイム待ち行列設定手段と、前記リアルタイムの計時終了割込要求発生に反応して該当タスクを起動するタスク起動手段とを備えた計算機システムのタスク待ち時間管理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は計算機システムにおいて、指定待ち時間経過後に起動させるタスク実行要求に対して、指定された正確なタイミングで該当タスクを起動させる計算機システムのタスク待ち時間管理装置に関する。

## 【0002】

【従来の技術】 システムバスに対して、各種情報演算処理を実行する CPU（中央演算処理装置）、OS（オペレーティング・システム）やアプリケーションプログラム等を記憶する ROM、各種可変データを記憶する RAM、外部機器に対するデータ送受信を行う入出力ポート等を備えた計算機システムにおいては、アプリケーションプログラムに基づいて情報処理業務を実行している過程で、多数のタスク実行要求が発生する。

【0003】 この多数のタスク実行要求のなかには、タスク実行要求が発生すると即座に該当タスクを実行する通常のタスク実行要求の他に、現在時刻から指定した待ち時間  $T_s$  経過後に該当タスクを実行するためのタスク実行要求もある。

【0004】 したがって、このような待ち時間  $T_s$  を指定したタスク実行要求に対応するために、上述した計算機システム内に、規定周期  $T_o$  で定期割込要求を前記 CPU へ印加するシステムタイマを設け、このシステムタイマの待ち行列に前記指定待ち時間  $T_s$  を設定する。そして、CPU はシステムタイマからの定期割込要求が発

2

生する毎に、システムタイマ待ち行列に設定された待ち時間  $T_s$  の計時が終了しているか否かを判断して、終了していれば、該当タスクを起動する。

## 【0005】

【発明が解決しようとする課題】 しかしながら、上述したようなタスク待ち時間管理手法においては、まだ改良すべき次のような画題があった。すなわち、上述した手法によると、指定された待ち時間  $T_s$  はシステムタイマの定期割込要求の数、すなわちスタック数で計時される。待ち時間  $T_s$  の計時精度は規定周期  $T_o$  に依存する。したがって、待ち時間  $T_s$  の計時精度を向上させるためには、規定周期  $T_o$  を短くする必要がある。

【0006】 しかし、システムタイマから CPU に印加する定期割込要求の発生間隔を短くすると、CPU における上述した時間割込処理回数が増加して、CPU の処理負担が増大して、他の通常の処理能率が低下する懸念がある。

【0007】 本発明はこのような事情に鑑みてなされたものであり、タスク実行要求の指定待ち時間を計時するタイマを通常のシステムタイマの他に詳細時間を計時するリアルタイムとの 2 つのタイマで計時することによって、CPU に余分な割込処理負担をかけることなく、指定待ち時間が示す正確なタイミングでタスクを起動でき、待ち時間管理精度を大幅に向上できる計算機システムの待ち時間管理装置を提供することを目的とする。

## 【0008】

【課題を解決するための手段】 上記課題を解消するために本発明の待ち時間管理装置は、現在時刻から指定待ち時間経過後に起動させるタスク実行要求を待ち行列に設定して、この指定待ち時間経過後に該当タスクを起動させる計算機システムに適用される。

【0009】 そして、規定周期で定期割込要求を発生するシステムタイマと、現在時刻からシステムタイマの次の定期割込要求発生時刻までの端数時間を計時すると共に、リアルタイム待ち行列に設定された時間の計時が終了すると計時終了割込要求を発生するリアルタイムと、待ち時間を指定したタスク実行要求に応じて、この指定待ち時間から端数時間を差引いた時間を規定周期で除算して商と余り時間を求める除算手段と、除算手段にて得られた商を該当タスクの待ちスタック数としてシステムタイマ待ち行列に設定するシステムタイマ待ち行列設定手段と、システムタイマ待ち行列に設定されたスタック数に対応する数の規定周期経過後の定期割込要求発生に応じて、該当タスクの余り時間をリアルタイム待ち行列に設定するリアルタイム待ち行列設定手段と、リアルタイムの計時終了割込要求発生に反応して該当タスクを起動するタスク起動手段とを備えたものである。

## 【0010】

【作用】 このように構成されたタスク待ち時間管理装置において、待ち時間  $T_s$  を指定したタスク実行要求が発

(3)

特開平7-152588

3

4

生すると、現在時刻から次の定期割込要求発生時刻までの端数時間 $T_r$ がリアルタイムから読取られる。したがって、待ち時間 $T_s$ からこの端数時間 $T_r$ を差引いた時間 $(T_s - T_r)$ が次の定期割込要求発生時刻から該当タスクを起動すべき時刻までの時間となる。

【0011】この時間 $(T_s - T_r)$ を規定周期 $T_o$ で除算して商 $N$ と余り時間 $n$ が算出される。この商 $N$ がシステムタイマ待ち行列にスタック数として設定される。したがって、スタック数 $N$ に対応する数の規定周期 $(N \cdot T_o)$ 経過後の定期割込要求発生に応じて、前記余り時間 $n$ をリアルタイム待ち行列に設定すれば、この余り時間 $n$ 経過後に該当タスクが起動する。すなわち、最初のタスク実行要求の発生時刻から $(N \cdot T_o + n = T_s)$ 経過後に該当タスクが起動する。

【0012】前記リアルタイムは余り時間 $n$ の計時が終了したタイミングで計時終了割込要求を出力するのみであるので、たとえこのリアルタイムの計時精度を向上させたとしても、CPUに対する割込要求の発生頻度が上昇することはない。CPUに対する割込要求は、システムタイマからの規定周期 $T_o$ 経過毎の定期割込要求とリアルタイムからの1つの計時終了割込要求のみであるので、CPUの割込処理負担が大幅に増大することはない。

【0013】

【実施例】以下本発明の一実施例を図面を用いて説明する。図1は実施例のタスク待ち時間管理装置が組込まれた計算機システムの概略構成を示すブロック図である。

【0014】システムバス1に対して、各種情報処理を実行するCPU2、OSやアプリケーションプログラム等の固定データを記憶するROM3、各種可変データを記憶する主メモリとしてのRAM4、外部機器に対するデータ送受信を行う入出力ポート5、システムタイマ6及びリアルタイム7等が接続されている。

【0015】システムタイマ6は、例えば100ms等の予め定められた規定周期 $T_o$ 経過する毎にCPU2の一方の割込端子INT1へ定期割込要求 $R_s$ を送出する。一方、リアルタイム7は、起動された時刻から図2(b)に示すリアルタイム待ち行列8における先頭の領域8aに設定されている各時間 $t_1$ (余り時間 $n$ )が到来すると計時終了割込要求 $R_r$ をCPU2の他方の割込端子INT2へ送出的。次に、その計時終了割込要求 $R_r$ の発生時刻から2番目の領域8aに設定されている時間 $t_2$ が経過すると、2回目の計時終了割込要求 $R_r$ をCPU2へ送出的。このように、リアルタイム7は各領域8aに設定された各時間の計時が終了する毎に順次CPU2へ計時終了割込要求 $R_r$ を送出する。

【0016】また、このリアルタイム7は、常時、現在時刻 $t$ から前記システムタイマ6が出力する次の周期割込要求 $R_s$ の発生時刻までの端数時間 $T_r$ を計時している。図2(a)はCPU2内又はRAM4内に形成され

るシステムタイマ待ち行列9の構成を示す図である。このシステムタイマ待ち行列9内には1番目、2番目、3番目、…と領域番号が付された複数の領域9aが形成されている。

【0017】同様に、図2(b)はCPU2内又はRAM4内に形成される前述したリアルタイム待ち行列8の構成を示す図である。このリアルタイム待ち行列8内には1番目、2番目、3番目、…と領域番号が付された複数の領域8aが形成されている。

【0018】このような構成の計算機システムにおいて、CPU2は、アプリケーションプログラムに従った通常の業務処理過程において、待ち時間 $T_s$ を指定したタスク実行要求が発生すると、図3に示すタスク割付処理を実行するようにプログラム構成されている。

【0019】図3の流れ図が開始されると、P(プログラムステップ)1において、生じたタスク実行要求の指定待ち時間 $T_s$ とシステムタイマ6の規定周期 $T_o$ との大小を比較する。指定待ち時間 $T_s$ が大きい場合は、このタスク実行要求は現在時刻から1規定周期 $T_o$ 以降に実行されるので、リアルタイム7にて計時されている現在時刻から次の定期割込要求 $R_s$ の発生時刻までの端数時間 $T_r$ を読取る(P2)。

【0020】次に、指定待ち時間 $T_s$ から端数時間 $T_r$ を差引いた時間 $(T_s - T_r)$ を規定周期 $T_o$ で除算して、商 $N$ と余り時間 $n$ を求める(P3)。そして、商 $N$ 、余り時間 $n$ 及び該当タスク名をタスク待ち行列9における空き領域の先頭領域9aに設定する(P4)。なお、商 $N$ はスタック数として該当領域8aに設定される。そして、システムタイマ待ち行列9内において複数の領域9aにスタック数 $N$ (商)、余り時間 $n$ 、タスク名が設定されていた場合は、各領域9aをスタック数 $N$ (商)の小さい順番に並べ替える(P5)。その結果、1番目の領域9aには最も小さいスタック数 $N$ (商)のタスク実行要求が設定されることになる。

【0021】P1において、指定待ち時間 $T_s$ が規定周期 $T_o$ より短い場合は、このタスク実行要求は現在時刻から1規定周期 $T_o$ 以内に実行されるので、P6へ進み、直ちに、この指定待ち時間 $T_s$ に対する図5に示すリアルタイム待ち行列設定処理を実行する。

【0022】また、前記CPU2は割込端子INT1にシステムタイマ6から定期割込要求 $R_s$ が入力すると図4に示す定期割込処理を実行する。図4の流れ図が開始されると、Q1にて、システムタイマ待ち行列9における各領域9aのスタック数 $N$ の値を調べる。Q2にて、 $N=0$ のスタック数のタスク実行要求が存在しなければ、Q3にて、システムタイマ待ち行列9を構成する全部の領域9aのスタック数 $N$ を1だけ減算する( $N=N-1$ )。

【0023】Q2にて、 $N=0$ のスタック数のタスク実行要求が存在すると、Q4にて、該当タスクに対する図

(4)

特開平 7-152588

5

5に示すリアルタイム待ち行列設定処理を実行する。その後、 $N=0$ のタスク実行要求を該当領域9aからクリアする。そして、Q6にて、システムタイム待ち行列9を構成する全部の領域9aのスタック数 $N$ を1だけ減算する( $N=N-1$ )。

【0024】図5は図3のP6又は図4のQ4にて起動されるリアルタイム待ち行列設定処理を示す流れ図である。流れ図が開始されると、S1にてリアルタイム待ち行列8の領域8aに設定すべき時間 $t$ を決定する。図3のP6にて起動された場合は、指定待ち時間 $T_s > \text{規定周期 } T_o$ であるので、設定すべき時間 $t$ は指定待ち時間 $T_s$ そのものとなる。

【0025】 $t = T_s$

また、図4のQ4にて起動された場合は、この処理が開始された時点においては、タスク実行要求発生時刻から既に( $N \cdot T_o$ )の時間が経過しているため、余り時間 $n$ を設定すべき時間 $t$ とする。

【0026】 $t = n$

次に、リアルタイム待ち行列8の各領域8aに未実行のタスク実行要求が存在するか否かを調べる(S2)。存在しなければ、S3へ進み、今回決定した時間 $t$ 及び該当タスク名を先頭の領域8aに設定して、リアルタイム7を起動させる(S5)。

【0027】リアルタイム7は起動されて先頭の領域8aに設定された時間 $t$ の計時が終了するとCPU2の割込端子INT2に計時終了割込要求 $R_r$ を送出する。その結果、CPU2は該当時間 $t$ の領域8aのタスク名のタスクを実行(起動)する。なお、該当タスクが起動されると、このタスク名及び時間 $t$ を領域8aからクリアする。

【0028】S2において、リアルタイム待ち行列8の領域8aに既に他のタスク実行要求の時間 $T$ 及びタスク名が設定されていれば、S6にて各領域8aに設定されている他の各時間 $T$ を読出する。

【0029】一つの領域8aに設定されている時間 $T$ と今回設定しようとする時間 $t$ との大小を比較する(S7)。今回の時間 $t$ が既に設定されている時間 $T$ より大きい場合は、今回のタスクは既に設定されている時間 $T$ のタスクが実行された時刻から( $t-T$ )の時間経過後に実行されると判断する。

【0030】S9にて、他の領域8aにタスク実行要求が設定されていなければ、S3へ進み、今回の時間 $t = (t-T)$ 及びタスク名を新規の領域8aへ設定する。今回は、既にリアルタイム7は計時動作中であるので(S4)、リアルタイム7を起動することなくそのまま今回の流れ図を終了する。

【0031】S9において、さらに、別の領域8aにタスク実行要求が設定されていた場合は、S10にて該当領域8aの時間 $T$ を読取って、S7へ戻り、今回の時間 $t (= t-T)$ との大小を比較する。

6

【0032】今回の時間 $t$ が既に設定されている時間 $T$ より小さい場合は、この時間 $T$ に対応するタスクが実行される前に今回のタスクを実行する必要がある。したがって、S11へ進み、既に設定されている他のタスクに対応する時間 $T$ を $T = (T-t)$ に変更する。すなわち、既に設定されている時間 $T$ のタスクを今回設定する時間 $t$ のタスクの起動時刻から( $T-t$ )の時間経過後に起動するように設定する。

【0033】このように構成されたタスク待ち時間管理装置における実際の動作をタイムチャートを用いて説明する。図6に示すように、100msの規定周期 $T_o$ で定期割込要求 $R_s$ が発生している状態において、待ち時間 $T_s = 200\text{ms}$ のタスク実行要求が一つの定期割込要求 $R_s$ 発生時刻から30ms経過後に発生したとする。

【0034】この場合、端数時間 $T_r$ は70msとなる。また、待ち時間 $T_s$ から端数時間 $T_r$ を差引いた時間( $T_s - T_r$ )は130msとなる。そして、この時間( $T_s - T_r$ )を規定周期 $T_o$ で除算すると、商 $N$ (スタック数)は1となり、余り時間 $n$ は30msとなる。

【0035】したがって、スタック数 $N=1$ 、余り時間 $n=30\text{ms}$ 及びタスク名がシステムタイム待ち行列9の一つの領域9aに設定される。そして、現在時刻から2回目の定期割込要求 $R_s$ がCPU2へ送出されたタイミングで、余り時間 $n$ が時間 $t$ としてリアルタイム待ち行列8の一つの領域8aに設定される。

【0036】この場合、リアルタイム待ち行列8に他のタスク実行要求が設定されていない場合は、即座にリアルタイム7が起動される。そして、設定した時間 $t$ (余り時間 $n$ )が経過するとCPU2に対して計時終了割込要求 $R_r$ が送出され、該当タスクが起動される。

【0037】また、既に他のタスク実行要求がリアルタイム待ち行列8に設定されていた場合は、既に設定されている他の時間 $T$ との比較を実行しながら、今回の時刻 $t$ を所定領域8aに設定する。

【0038】また、指定待ち時間 $T_s$ が規定周期 $T_o$ より短い場合は、直ちに指定待ち時間 $T_s$ がリアルタイム待ち行列8の領域8aに時間 $t$ として設定されるが、この場合、既に他のタスク実行要求が他の領域8aに設定されていた場合における今回のタスクに対応する時間 $t$ の設定手順を図7を用いて説明する。

【0039】図7は、既にタスクA、タスクB及びタスクCの3つのタスクの各時間 $T_a$ 、 $T_b$ 、 $T_c$ が各領域8aにそれぞれ設定されている状態において、別のタスクX、タスクY及びタスクZの3つのタスクにおける各待ち時間 $T_x$ 、 $T_y$ 、 $T_z$ ( $T_x < T_y < T_z$ )をそれぞれ各領域8aに設定する場合を説明するための図である。

50 【0040】なお、タスクAの前には他のタスクの実行

(5)

特開平7-152588

7

要求が存在しなくて、タスクAの実行要求発生時刻からリアルタイム7が起動され、時間 $T_a$ 経過後にタスクAが起動され、その後時間 $T_b$ 経過後にタスクBが起動され、その後時間 $T_c$ 経過後にタスクCが起動される。

【0041】この初期状態においては、図8(a)に示すように、リアルタイム待ち行列8の1番目から3番目までの各領域8aにはそれぞれ各タスクA～Cに対応する各時間 $T_a$ 、 $T_b$ 、 $T_c$ が設定されている。

(a) そして、タスクXの時間 $T_x$ を設定する場合は、図7より

$T_x < t_a$  (現在時刻から次のタスクAが起動するまでの時間)

であるので、図8(b)に示すように、1番目の領域8aに今回のタスクXに対する時間 $T_x$ が設定され、2番目の領域8aにタスクAの時間 $T_a$ を訂正した時間( $t_a - T_x$ )が設定される。そして、3番目及び4番目の各領域8aにはそれぞれ上位番目の領域8aからシフトされた各時間 $T_b$ 、 $T_c$ が設定される。

(b) また、タスクYの時間 $T_y$ を設定する場合は、図7より

$t_a < T_y < t_a + T_b$

であるので、図8(c)に示すように、1番目の領域8aのタスクAの時間 $T_a$ は変化せずに、2番目の領域8aにタスクYに対する時間( $T_y + t_a$ )が追加される。3番目の領域8aにはタスクBの時間 $T_b$ を変更した時間 $[T_b - (T_y - t_a)]$ が設定される。4番目の領域8aには3番目の領域8aからシフトしたタスクCに対する時間 $T_c$ が設定される。

(c) さらに、タスクZの時間 $T_z$ を設定する場合は、図7より

$t_a + T_b < T_z < t_a + T_b + T_c$

であるので、図8(d)に示すように、1番目の領域8a及び2番目の領域8aの各時間 $T_a$ 、 $T_b$ は変化せずに、3番目の領域8aにタスクZに対する時間 $[T_z - (t_a + T_b)]$ が追加される。4番目の領域にはタスクCの時間 $T_c$ を変更した時間 $[T_c - (T_z - t_a - T_b)]$ が設定される。

【0042】そして、リアルタイム7は、リアルタイム待ち行列8の各番目の領域8aの計時が終了した各タイミングでそれぞれ計時終了割込要求 $R_r$ をCPU2へ送出すると共に、次の番目の領域8aの時間の計時を開始する。

【0043】よって、各タスクA～C、X～Zの実行要求はそれぞれ指定された正しい待ち時間経過後に起動される。このように構成されたタスク待ち時間管理装置であれば、待ち時間 $T_s$ を指定したタスク実行要求が発生すると、まず、指定待ち時間 $T_s$ と規定周期 $T_o$ とが比較され、指定待ち時間 $T_s$ が規定周期 $T_o$ より短い場合は、即座にこの待ち時間 $T_s$ がリアルタイム7のリアルタイム待ち時間行列8に設定される。

8

【0044】また、指定待ち時間 $T_s$ が規定周期 $T_o$ より長い場合には、現在時刻から次の定期割込要求 $R_s$ の発生時刻までの端数時間 $T_r$ がリアルタイム7から読取られ、この待ち時間 $T_s$ からこの端数時間 $T_r$ を差引いた時間( $T_s - T_r$ )を規定周期 $T_o$ で除算して商 $N$ と余り時間 $n$ が算出される。この商 $N$ がシステムタイム待ち行列9の領域9aにステック数として設定される。

【0045】したがって、ステック数 $N$ に対応する数の規定周期( $N \cdot T_o$ )経過後の定期割込要求 $R_s$ 発生に応じて、余り時間 $n$ がリアルタイム待ち行列8の領域8aに時間 $t$ として設定され、この時間 $t$ 経過後に該当タスクが起動する。

【0046】このように、リアルタイム7は余り時間 $n$ に対応する時間 $t$ の計時が終了したタイミングで計時終了割込要求 $R_r$ を出力するのみであるので、たとえこのリアルタイム7の計時精度を向上させたとしても、CPU2に対するシステムタイム6及びリアルタイム7からの各割込要求 $R_s$ 、 $R_r$ の発生頻度が上昇することはない。したがって、CPU2の割込処理負担が大幅に増大することはない。

【0047】また、結果的に、待ち時間を指定したタスク実行要求における指定待ち時間 $T_s$ をシステムタイム6及びリアルタイム7との2つのタイムで分割して計時している。すなわち、100ms等の大きな単位をシステムタイム6で計時し、100ms以内の例えば1ms間隔の細かい単位をリアルタイム7で計時している。リアルタイム7はリアルタイム待ち行列8の各領域8aにタスク実行要求がなくなれば、計時値を0にクリして、起動待ちになる。したがって、リアルタイム7の計時範囲をそんなに大きく設定する必要がないので、クロック周期を短くして設定された時間の計時精度を容易に向上できる。

【0048】

【発明の効果】以上説明したように本発明のタスク待ち時間管理装置によれば、タスク実行要求の指定待ち時間を計時するタイムを、規定周期で定期割込要求を出力する通常システムタイムの他に、待ち時間から端数時間を差し引いた時間を規定周期で除算した場合における余り時間を詳細に計時するリアルタイムとの2つのタイムで計時している。したがって、CPUに余分な割込処理負担が加わることなく、各指定待ち時間が示す正確なタイミングで各タスクを起動でき、待ち時間管理精度を大幅に向上できる。

【図面の簡単な説明】

【図1】 本発明の一実施例に係わるタスク待ち時間管理装置が組込まれた計算機システムの概略構成を示すブロック図

【図2】 同実施例タスク待ち時間管理装置に組込まれたシステムタイム待ち行列及びリアルタイム待ち行列を示す図

- 9
- 【図3】 同実施例装置のタスク割付処理を示す流れ図

【図4】 同実施例装置の定期割付処理を示す流れ図

【図5】 同実施例装置のリアルタイム待ち行列設定処理を示す流れ図

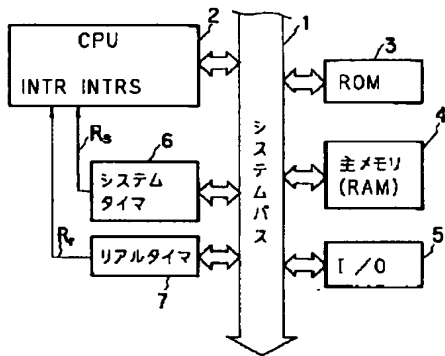
【図6】 同実施例装置の動作を示すタイムチャート

【図7】 同じく同実施例装置の動作を示すタイムチャート

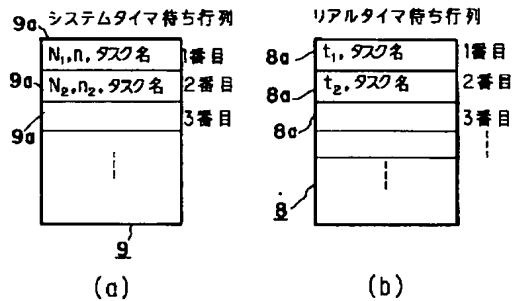
(6)

特開平7-152588

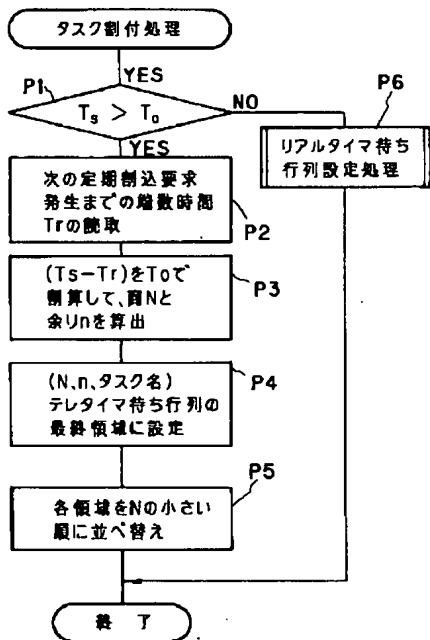
【図1】



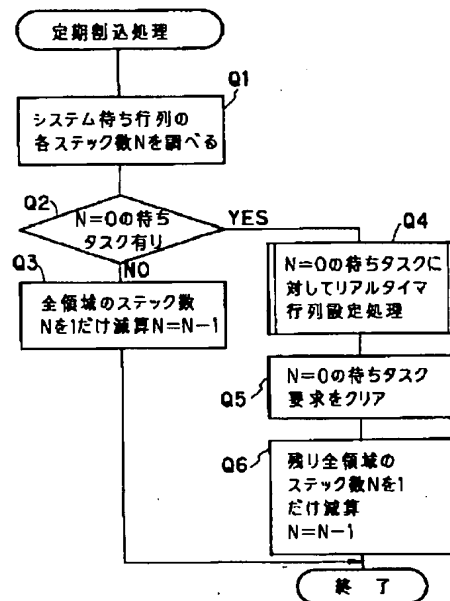
【図2】



【図3】



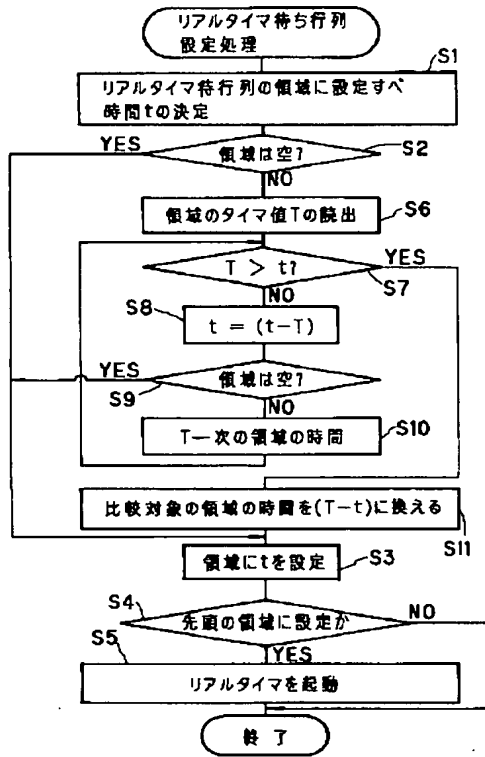
【図4】



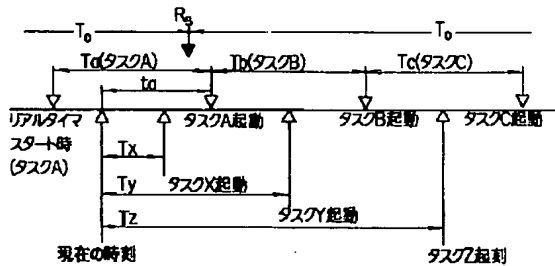
(7)

特開平 7-152588

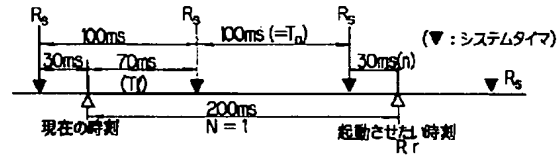
【図5】



【図7】



【図6】



【図8】

